

**Amendment to the Claims:**

Without prejudice, this listing of the claims replaces all prior versions and listings of the claims in the present application:

**Listing of Claims:**

1. (Currently Amended) A method of managing visibility of GUI components in an application comprising:

providing a user interface of a visibility manager for selection, for each of a plurality of profiles, of one of a plurality of visibility states for each of at least a subset of the GUI components;

initializing the application;

invoking the ~~[[a]]~~ visibility manager; and

displaying a user interface of the application wherein display of the GUI components are determined by the visibility manager based on the selected visibility states of an applied one of the plurality of profiles.

2. (Currently Amended) The method as recited in claim 1 wherein invoking the visibility manager comprises:

reading ~~one or more~~ the plurality of profiles;

processing the ~~one or more~~ plurality of profiles;

reading and processing a user configuration based on the ~~one or more~~ plurality of profiles; and

activating the applied ~~a particular profile of the one or more~~ profiles.

3. (Currently Amended) The method as recited in claim 2 wherein invoking the visibility manager further comprises:

revising the user interface based on the applied ~~particular profile of the one or more~~ profiles; and

displaying the user interface.

4. (Currently Amended) The method as recited in claim 3 wherein initializing the application comprises:

- starting the application;
- building the user interface with all of the GUI components visible; and
- calling the visibility manager after the applied ~~particular~~ profile of ~~the one or more~~ profiles is activated.

5. (Currently Amended) The method as recited in claim 2 wherein activating the applied ~~particular~~ profile of ~~the one or more~~ profiles comprises:

- selecting an identification of a particular GUI component;
- locating the identification in a mapping table;
- checking a state of the particular GUI component;
- comparing the state to the applied ~~particular~~ profile of ~~the one or more~~ profiles;
- changing the state if not in agreement with the applied ~~particular~~ profile of ~~the one or more~~ profiles; and
- repeating locating the identification, checking the state of the particular GUI component, comparing the state and changing the state for any remaining additional identifications of additional GUI components.

6. (Original) The method as recited in claim 5 wherein the state is visible or not visible.

7. (Currently Amended) The method as recited in claim 5 wherein the mapping table comprises a plurality of identifications of GUI components and a corresponding plurality of references to [[Java]] objects of an object-oriented and platform independent programming language.

8. (Currently Amended) The method as recited in claim 1 further comprising:  
invoking the visibility manager by:

- reading ~~one or more~~ the plurality of profiles,
- processing the ~~one or more~~ plurality of profiles,
- reading and processing a user configuration based on the ~~one or more~~ plurality of profiles,

- activating the applied ~~a particular~~ profile ~~of the one or more profiles~~ by:

- selecting an identification of a particular GUI component,
  - locating the identification in a mapping table,
  - checking a state of the particular GUI component,
  - changing the state if not in agreement with the applied ~~particular~~

- profile ~~of the one or more profiles~~, and

- repeating locating the identification, checking the state of the particular GUI component, comparing the state and changing the state for any remaining additional identifications of additional GUI components,

- revising the user interface based on the applied ~~particular~~ profile ~~of the one or more profiles~~, and

- displaying the user interface;

initializing the application by:

- starting the application,

- building the user interface with all of the GUI components visible, and

- calling the visibility manager after the applied ~~particular~~ profile ~~of the one or more profiles~~ is activated.

9. (Currently Amended) A system for managing visibility of GUI components in an application comprising:

a processor configured to provide:

a user interface module of the application, including the GUI components;

[[and]]

a visibility manager that determines which GUI components are visible; and

a user interface of the visibility manager via which to receive, for one or more profiles, input of respective selections of visibility states of at least a subset of the GUI components, the visibility manager determining which GUI components are visible based on the selections.

10. (Currently Amended) The system as recited in claim 9, wherein the visibility manager to determine which GUI components to display by, and the visibility manager further to:

read the one or more profiles upon initialization of the application;

process the one or more profiles;

read and process a user configuration based on the one or more profiles; and

activate a particular profile of the one or more profiles.

11. (Original) The system as recited in claim 10 wherein the user interface module to:

build the user interface with all of the GUI components visible;

the visibility manager to:

call the visibility manager after the particular profile of the one or more profiles is activated;

revise the user interface based on the particular profile of the one or more profiles; and

the user interface module further to:

display the user interface.

12. (Original) The system as recited in claim 10 wherein the visibility manager to:  
select an identification of a particular GUI component;  
locate the identification in a mapping table;  
check a state of the particular GUI component;  
compare the state to the particular profile of the one or more profiles; and  
change the state if not in agreement with the particular profile of the one or more profiles.

13. (Original) The system as recited in claim 12 wherein activating the particular profile of the one or more profiles further comprises repeating locating the identification, checking the state of the particular GUI component, comparing the state and changing the state for any remaining additional identifications of additional GUI components.

14. (Original) The system as recited in claim 12 wherein the state is visible or not visible.

15. (Currently Amended) The system as recited in claim 12 wherein the mapping table comprises a plurality of identifications of GUI components and a corresponding plurality of references to [[Java]] objects of an object-oriented and platform independent programming language.

16. (Currently Amended) The system as recited in claim 9 further wherein the visibility manager to determine which GUI components to display by, and the visibility manager to:

read the one or more profiles upon initialization of the application,  
process the one or more profiles,  
read and process a user configuration based on the one or more profiles,  
activate a particular profile of the one or more profiles by:  
select an identification of a particular GUI component,  
locate the identification in a mapping table,  
check a state of the particular GUI component,  
compare the state to the particular profile of the one or more profiles,  
and

change the state if not in agreement with the particular profile of the one or more profiles,  
start the application,  
build the user interface with all of the GUI components visible,  
call the visibility manager after the particular profile of the one or more profiles is activated,  
revise the user interface based on the particular profile of the one or more profiles, and  
display the user interface.

17. (Currently Amended) A system comprising:  
a processor configured to:

use a visibility manager data structure for use in managing visibility of GUI components in a user interface of an application, the visibility manager data structure comprising a mapping table, one or more profiles and a user configuration identifying which of the one or more profiles is to be applied; and

provide a user interface of a visibility manager via which to receive for the one or more profiles input of respective selections of visibility states of at least a subset of the GUI components, the visibility of the GUI components determined in accordance with the visibility state selections of the identified profile.

Claim 18. (Canceled).

19. (Currently Amended) The ~~visibility manager data structure~~ system as recited in claim [[18]] 17, wherein

the visibility manager to determine which GUI components to display, and the visibility manager further to:

read the one or more profiles upon initialization of the application;  
process the one or more profiles;  
read and process the user configuration based on the one or more profiles; and  
activate ~~a particular~~ the identified profile of the one or more profiles based upon the mapping table.

20. (Currently Amended) The ~~visibility manager data structure~~ system as recited in claim 19 wherein

the visibility manager to further determine which GUI components are visible in the application, and the visibility manager further to:

revise the user interface based on the ~~particular~~ identified profile of the one or more profiles; and

display a user interface.

21. (Currently Amended) The ~~visibility manager data structure~~ system as recited in claim 19 wherein the visibility manager to:

select an identification of a particular GUI component;

locate the identification in ~~[[a]]~~ the mapping table;

check a state of a specific GUI component in the application;

compare the state to the ~~particular~~ identified profile of the one or more of profiles; and

change the state if not in agreement with the ~~particular~~ identified profile of the one or more profiles; and

~~wherein the visibility manager to select the identification, locate the identification, check the state of the particular GUI component, compare the state and change the state if not in agreement with the particular profile of the one or more profiles.~~

22. (Currently Amended) The ~~visibility manager data structure~~ system as recited in claim 21 wherein the state is visible or not visible.

23. (Currently Amended) The ~~visibility manager data structure~~ system as recited in claim 17 wherein the mapping table comprises a plurality of identifications of GUI components and a corresponding plurality of references to ~~[[Java]]~~ objects of an object-oriented and platform independent programming language.

24. (Currently Amended) The ~~visibility manager data structure~~ system as recited in claim 18 wherein

the visibility manager to determine which GUI components to display, and the visibility manager further to:

- read the one or more profiles upon initialization of the application;
- process the one or more profiles;
- read and process the user configuration based on the one or more profiles;
- activate the identified ~~a particular~~ profile of the one or more profiles based

upon the mapping table by:

- select an identification of a particular GUI component,
- locate the identification in ~~[[a]]~~ the mapping table,
- check a state of a specific GUI component in the application,
- compare the state to the identified ~~particular~~ profile of the one or more of profiles, and
- change the state if not in agreement with the identified ~~particular~~ profile of the one or more profiles;
- revise the user interface based on the activated profile, and
- display a user interface.

25. (Currently Amended) A hardware-implemented computer-readable medium embodying instructions, which, when executed by a processor, cause the processor to perform a method, the method comprising:

providing a user interface of a visibility manager via which to receive, for one or more profiles, input of respective selections of visibility states of at least a subset of GUI components;

- initializing an application;
- invoking a visibility manager; and
- displaying a user interface of the application wherein display of the GUI components are determined by the visibility manager based on the visibility state selections.



26. (Currently Amended) The medium as recited in claim 25 wherein invoking the visibility manager comprises:

- reading the one or more profiles;
- processing the one or more profiles;
- reading and processing a user configuration based on the one or more profiles;
- activating a particular profile of the one or more profiles;
- revising the user interface based on the particular profile of the one or more profiles;

and

- displaying the user interface.

27. (Original) The medium as recited in claim 26 wherein initializing the application comprises:

- starting the application;
- building the user interface with all of the GUI components visible; and
- calling the visibility manager after the particular profile of the one or more profiles is activated.

28. (Currently Amended) The medium as recited in claim 26 wherein activating the particular profile of the one or more profiles comprises:

- selecting an identification of a particular GUI component;
- locating the identification in a mapping table;
- checking a state of the particular GUI component;
- comparing the state to the particular profile of the one or more profiles; and
- changing the state if not in agreement with the particular profile of the one or more profiles.

29. (Original) The medium as recited in claim 26 further comprising:

- repeating locating the identification, checking the state of the particular GUI component, comparing the state and changing the state for any remaining additional identifications of additional GUI components.

30. (Original) The medium as recited in claim 28 wherein the state is visible or not visible.

31. (Currently Amended) The medium as recited in claim 28 wherein the mapping table comprises a plurality of identifications of GUI components and a corresponding plurality of references to [[Java]] objects of an object-oriented and platform independent programming language.

32. (Currently Amended) The medium as recited in claim 25 further comprising:  
invoking the visibility manager by:

- reading the one or more profiles,
- processing the one or more profiles,
- reading and processing a user configuration based on the one or more profiles,
- activating a particular profile of the one or more profiles by:

- selecting an identification of a particular GUI component,
  - locating the identification in a mapping table,
  - checking a state of the particular GUI component,
  - changing the state if not in agreement with the particular profile of the one or more profiles, and

- repeating locating the identification, checking the state of the particular GUI component, comparing the state and changing the state for any remaining additional identifications of additional GUI components,
  - revising the user interface based on the particular profile of the one or more

profiles, and

- displaying the user interface; and

initializing the application by:

- starting the application,
  - building the user interface with all of the GUI components visible, and
  - calling the visibility manager after the profile is activated.

33. (Currently Amended) A system for managing visibility of GUI components in an application comprising:

first means for interfacing with a user, the means for interfacing ~~providing including~~ the GUI components for display; and

means for determining which GUI components are visible, the means for determining including second means for interfacing with a user, the second means for interfacing providing for receipt, for one or more profiles, of respective user selections of visibility states of at least a subset of the GUI components, the selections used for the determination of which GUI components are visible.

34. (Currently Amended) The system as recited in claim 33 wherein the means for determining includes:

means for reading the one or more profiles upon initialization of the application;

means for processing the one or more profiles;

means for reading and processing a user configuration based on the one or more profiles; and

means for activating a particular profile of the one or more profiles.

35. (Currently Amended) The system as recited in claim 34 further comprising:

means for starting the application;

wherein:

the first means for interfacing with the user initially build for building the user interface with all of the GUI components visible;

the means for determining which GUI components are visible performs the determination after the particular profile of the one or more profiles is activated; and

the first means for interfacing with the user displays ~~displaying~~ the user interface.